JOINT
LOGISTICS
COMMANDERS

SAN ANTONIO I
SOFTWARE WORKSHOP

EXECUTIVE SUMMARY

DECEMBER 1991

"DOD SOFTWARE FOR THE 1990s"

28 JANUARY - 1 FEBRUARY 1991

(THIS IS NOT AN APPROVED JLC DOCUMENT)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Executive Summary

4. TITLE AND SUBTITLE
San Antonio I Software Workshop
Executive Summary

5. FUNDING NUMBERS
N/A

6. AUTHOR(S)
Joint Logistics Commanders Joint Policy Coordinating Group on Computer Resources Management, Computer Software Management Subgroup

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
See Item 9

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Commander Space and Naval Warfare Systems Command
Code 22412
Attn: Dr Raghu Singh
Washington, DC 20363-5100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The Action Plan is not available for secondary distribution

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited Distribution

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

San Antonio I was the fifth in a series of workshops focusing on relevant software acquisition and support issues pertinent to Mission Critical Computer Resources (MCCR). The previous workshops (Monterey I, Monterey II, Orlando I and Orlando II) were very instrumental in identifying issues that should be addressed in Department of Defense (DoD) standards for the development of mission critical systems. The central theme of San Antonio I was "DoD Software for the 1990s". Workshop selectees were assigned to one of seven panels. Each panel was assigned one particular problem area and tasked with developing solutions. The panels' conclusions reinforced the fact that more joint efforts are needed among the Services and between DoD and industry.

This Executive Summary contains summaries of the final reports of each of the panels.

See San Antonio I Software Workshop Proceedings for the final report of each panel.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Software, metrics, DOD-STD-2167A, DOD-STD-2168, security, configuration management, reusability, Ada secondary standards	29
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

JOINT LOGISTICS COMMANDERS FIFTH SOFTWARE WORKSHOP

SAN ANTONIO I

VOLUME I

EXECUTIVE SUMMARY

DECEMBER 11, 1991

PRODUCED FOR THE
JOINT LOGISTICS COMMANDERS JOINT POLICY COORDINATING GROUP
ON COMPUTER RESOURCES MANAGEMENT

THIS DOCUMENT WAS PRODUCED BY THE
COMPUTER SOFTWARE MANAGEMENT SUBGROUP

(INTENTIONAL BLANK)

FOREWORD

San Antonio I was the fifth in a series of workshops focusing on relevant software acquisition and support issues pertinent to Mission Critical Computer Resources (MCCR). The previous workshops (Monterey I, Monterey II, Orlando I, and Orlando II) were very instrumental in identifying issues that should be addressed in Department of Defense (DoD) standards for the development of mission critical systems. The central theme of San Antonio I was "DoD Software for the 1990s". Workshop selectees were assigned to one of seven panels. Each panel was assigned one particular problem area and tasked with developing solutions. The panels' conclusions reinforced the fact that more joint efforts are needed among the Services and between DoD and industry.

The panels addressed the following seven issues:

I. Software Metrics Implementation
II. DOD-STD-2167A and DIDs: Lessons Learned/Issues
III. DOD-STD-2168: Lessons Learned/Issues
IV. Computer Security/Software Integrity
V. Software Configuration Management
VI. Software Reusability
VII. Ada Secondary Standards

This Executive Summary contains summaries of the final reports of each of these panels. The Workshop Proceedings and Workshop Action Plan have been published as separate volumes.

Any questions concerning this material may be forwarded to:

Commander, Space and Naval Warfare Systems Command
Code 22412
Attn: Dr. Raghu Singh
Washington, DC 20363-5100

(INTENTIONAL BLANK)

TABLE OF CONTENTS

1.0 INTRODUCTORY MATERIAL

1.1 INTRODUCTION

The Joint Logistics Commanders (JLC) Joint Policy Coordinating Group on Computer Resources Management (JPCG-CRM) chartered the Computer Software Management (CSM) Subgroup in January 1980. The CSM Subgroup that organized the SA-I Workshop consisted of:

Maj Dan Romano	Air Force Systems Command (Chair)
Dr. Raghu Singh	Space and Naval Warfare Systems Command
Douglas Gerritsen	Army Armament Research and Development Center
Donald Kosco	Air Force Logistics Command
Ralph Wootton	Marine Corps Tactical Systems Support Activity

The current CSM Subgroup members (charged with collecting, cataloging, analyzing, and reporting the recommendations from SA-I) are:

Dr. Raghu Singh	Space and Naval Warfare Systems Command (Chair)
Wayne Sherer	Army Armament Research and Development Center
Donald Kosco	Air Force Logistics Command
Capt Steve Coyne	Air Force Systems Command
Jacquelyn Nixon	Marine Corps Tactical Systems Support Activity

The JPCG-CRM organized five joint Government/industry workshops attended by computer resource professionals. Those software workshops were "Monterey I" (1979), "Monterey II" (1981), "Orlando I" (1983), "Orlando II" (1987), and "San Antonio I" (1991).

The primary purpose of the San Antonio I Software Workshop (SA-I) was to review current software management issues common to the Services and industry, and to make specific recommendations concerning these issues. SA-I focused on acquisition management and development of mission critical computer resources (MCCR). The central theme of the workshop was "DoD Software for the 1990s". SA-I identified areas offering significant cost reduction, improved system reliability, and streamlining the software acquisition and support processes.

Panel membership was limited to 12 participants, equally divided between Government and industry representatives. Panel discussions focused on the key issues within each topic area. Panel Co-Chairs were responsible for directing the discussions to the topics and questions which pertained to the accomplishment of panel objectives. Interaction between panels was encouraged to ensure that all aspects of computer software were thoroughly investigated.

Three final reports summarizing SA-I will be issued. They will be the Workshop Executive Summary, the Workshop Proceedings, and the Workshop Action Plan. The Executive Summary will capture in an

optimal way what the Workshop was about and what high priority recommendations each panel made. The Proceedings will include a report from each panel detailing their recommendations. It will contain background, justification, actions required, projected benefit, estimated cost, and schedule for implementation for each recommendation. The Action Plan includes a list of prioritized workshop recommendations and defines organizational responsibilities, resources, and implementation schedules for these recommendations.

1.2 BACKGROUND

Under the sponsorship of the JLC, the JPCG-CRM hosted the San Antonio I (SA-I) Defense-Industry Software Workshop on 28 January through 1 February 1991 in San Antonio, Texas. The theme of the workshop was "DoD Software for the 1990s." The objectives were:

- to review software management and engineering issues common to the military-industrial complex,
- to learn from past experience,
- to understand future software technology, and
- to make specific recommendations for addressing these issues and technology.

The JLC is a self-chartered body endorsed by the Secretary of Defense. Members of the JLC are the commanders of:

- Army Materiel Command,
- Air Force Systems Command,
- Air Force Logistics Command,
- and the Deputy Chief of Naval Operations (Logistics).

The United States Marine Corps and the Defense Logistics Agency have been granted the role of invited participants. The primary goal of the JLC is to improve military effectiveness by addressing and exploiting opportunities for joint service cooperative efforts. The JLC have been meeting regularly since 1966.

There are currently several groups under JLC supervision. Each addresses a major area of concern. One of these groups, the JPCG-CRM, specifically addresses joint DoD and industry issues and concerns related to computers--both hardware and software. Membership is comprised of representatives from the Army, the Navy, and the Air Force, with the Marine Corps and DLA sitting in as invited participants. The JPCG-CRM has been meeting regularly since 1977.

The JPCG-CRM previously conducted four joint Defense-Industry workshops to address software related problems in the Services. These workshops were: Monterey I (1979), Monterey II (1981), Orlando I (1983), and Orlando II (1987). As a result of the Monterey I workshop, crucial needs for the consolidation of

existing, numerous, and diverse software standards were met. The necessity for DoD-wide software standards which can be tailored was also realized. Subsequent workshops made several key recommendations to significantly improve the software standards and practices. Based on these recommendations, the JPCG-CRM published the key software standards and handbooks:

- DOD-STD-2167 (Defense System Software Development) and related Data Item Descriptions (DIDs) in June 1985;
- DOD-STD-2167A (Defense System Software Development) and related DIDs in February 1988;
- DOD-STD-2168 (Defense System Software Quality Program) and related DID in April 1988;
- MIL-HDBK-287 (Tailoring Guide for DOD-STD-2167A) in August 1989;
- MIL-HDBK-347 (MCCR Software Support) in May 1990; and
- MIL-HDBK-286 (Application Guide for DOD-STD-2168) in December 1990.

The aforementioned standards and handbooks were the first of their kind to address the software development process across DoD. They have been credited with reducing cost, improving product quality, and providing incentive to the software industry to invest in software engineering environments based on these standards. Of course, there were still unresolved issues to be addressed with their origins in the application of the standards on real-life projects and recent technological developments. For this reason, the JPCG-CRM conducted SA-I, the fifth workshop in the series, to address these issues in the same spirit as the Monterey and Orlando workshops.

The SA-I workshop was organized into seven panels to address:

(1) Software Metrics Implementation;
(2) DOD-STD-2167A--Lessons Learned/Issues;
(3) DOD-STD-2168--Lessons Learned/Issues;
(4) Computer Security/Software Integrity;
(5) Software Configuration Management;
(6) Software Reusability; and
(7) Ada Secondary Standards.

Each panel had both a Government Co-chair and an industry Co-chair. Panel membership, limited to 12 participants, was equally divided between Government and industry members.

The workshop generated 137 recommendations to improve software acquisition, development, engineering, and management. The summaries of the panel recommendations are provided in later sections of this Executive Summary. Workshop proceedings and detailed recommendations are covered in a separate report entitled "Proceedings of the San Antonio I Workshop." The JPCG-CRM will review these recommendations and develop an Action Plan to implement

them. The JPCG-CRM will assume responsibility for several of these recommendations, particularly those related to the software standards and handbooks. For other recommendations, appropriate agencies will be requested to implement them. The Action Plan will be published separately.

2.0 INDIVIDUAL PANEL SYNOPSES

2.1 PANEL I: SOFTWARE METRICS IMPLEMENTATION

2.1.1 Required Policy Changes

The highest priority recommendation of Panel I was to establish policy at the DoD level which requires the implementation of software metrics for all Mission Critical Computer Resources (MCCR) software development programs. Barriers to such a task include policy which does not require the use of metrics, and inexperience by the projected users in the capabilities and limitations of software metrics technology. The challenge for the JLC is clearly in the development of a software metrics implementation approach which not only defines what should be done, but also educates acquisition managers on how to do it. This approach should be based upon practical experience in the application of software metrics to real software development programs.

The resulting software metrics program must prove its value to each Program Manager (PM) who must implement it. As the value of metrics is established, the scope of its application will increase. The long term objective is to use software metrics the same as any other program management tool: to assess, manage, and control the development of the software program.

Software metrics is a structured process based upon quantitative measures of software development activities and associated software products. Software metrics, when integrated into the program management process, have proven to be effective in supporting the successful development of MCCR software systems. Implemented correctly, software metrics can be a primary tool in the management of development program cost, schedule, and technical performance.

The panel was in agreement that PMs should be directed to implement metrics as part of their program management process. It was also expected that the continuing value of a properly implemented metrics program would be substantiated as the program matured.

2.1.2 Core Set of Validated Metrics Recommended

Panel I recommended that the JLC define and establish a software metrics implementation approach applicable to MCCR software development programs which incorporates software metrics training, effective data collection and analysis procedures, and the application of a "core" metrics set for both the pre-award and

software development life cycle phases. Recommended software measures comprise a basic set of "core" measures. The core set of metrics is the minimum set required for all MCCR software development programs. These measures are defined as those required to support software development objectives and issues common to all MCCR software developments. Measures pertinent to software development progress, resource expenditure, product quality, and program stability are included. The detail of the definition of these metrics was held at a relatively high level for the purposes of the panel recommendations. This was to ensure consistency with the key concepts and associated implementation guidelines.

Application of specific measures cannot be arbitrary. There has to be a reason for the measurement. The JLC should generate an effective software development metrics approach which will serve as the basis for implementing software metrics on MCCR software development programs. This approach should be based upon actual software metrics application experience and incorporate those procedures and measures which have proven to be effective. This implementation approach should include the following:

- o software metrics training,
- o software metrics applications guidance,
- o software measures for application prior to contract award, and
- o software measures for application during software development.

These four requirements are essential to the implementation of an effective software metrics program. The development of the detailed guidance and materials to implement these requirements should be the responsibility of a technical metrics center to be created by and working under the direction of the JLC.

The following software metrics support issue identification and assessment in the above listed areas. They are recommended as the "core" metrics to be implemented by all MCCR software development programs. Due to the number of measures which support each issue, they have been separated into "primary" and "secondary" groups. The primary group is recommended for initial implementation based upon data availability and ease of interpretation. The secondary group, although still considered to be "core" metrics, are recommended for implementation at a subsequent point in time.

Primary metrics are: Work Unit Completion, Software Size Allocation, Software Schedule Performance, Software Cost Performance, Staffing Performance, Effort Performance, Software Development Productivity, Facility Utilization, Software Defects, Software Requirements Conformance, Requirements Size/Scope Growth, Software Size/Scope Growth, Project Stability, and Rework Effort.

Secondary metrics are: Software Development Productivity, Facility Utilization, and Software Requirements Conformance.

2.2 PANEL II: DOD-STD-2167A AND DIDs: LESSONS LEARNED/ISSUES

2.2.1 Great Improvement

DOD-STD-2167A and its DIDs were published in February 1988. DOD-STD-2167A establishes:

- a standard terminology,
- a standard set of deliverables to choose from,
- a standard set of reviews and audits to choose from, and
- a standard set of software management practices that may be imposed during software development.

The DIDs provide format and content requirements for the deliverables which display visible evidence of the software development, design, implementation, and verification processes. In the three years since their publication, the standard and DIDs have been tailored and implemented on a wide range of MCCR developments in all phases of the acquisition cycle. To aid in form fitting the standard and its DIDs to specific projects, a tailoring handbook MIL-HDBK-287 was prepared and published. An update to DOD-STD-2167A is planned for 1993.

DOD-STD-2167A has been well received by the acquisition community and contractors. Despite the shortcomings stated herein, the standard has many advantages and is certainly the best standard to use until a revision is provided. Government use of the standard has been largely limited to the procurement process.
It has not yet been extensively used in the Post Deployment Software Support (PDSS) arena. In addition, the standard's use by contractors has resulted in its shortcomings being exposed. Users of the standard have made their experience known for the sake of improving the standard. However, DOD-STD-2167A is still considered a very useful standard. Therefore, caution should be used in making changes to the standard to ensure that the next version will be a step forward.

2.2.2 Systems Interfaces and Security Issues

Panel II felt that the systems interface with software needed strengthening. Their primary recommendations included:

- strengthen systems engineering support for software by establishing a JLC System Engineering Managers group tasked to initiate the development of a new standard which formalizes the system engineering process and its interface with DOD-STD-2167A;

- o expand DOD-STD-2167A to include developing the software architecture and define the role of software engineering in interfacing the software with the remainder of the system; and
- o include a requirement to support systems engineering in the preparation of a draft "System User's Manual".

The Security Panel (Panel IV) presented "Lessons Learned Using DOD-STD-2167A" to Panel II Co-chairs. These considerations were then described to the full Panel II membership and the consensus was that software security should be upgraded to a major issue in DOD-STD-2167A. It was envisioned that this goal could be accomplished by adding references to security requirements in the standard and DIDs wherever appropriate.

2.2.3 Changes to Definitions, Requirements, and Explanations

A number of recommendations for changes, amplifications, and clarifications to the standard, DIDs, and handbook were made for ease of use and to support a wider range of needs. In particular, it was recommended that the terms "states", "modes", and "capabilities" be defined and their usage in the DIDs further explained.

A large number of the submitted "lessons learned" were on documentation. Several recommendations arose from PDSS experience and entailed changing the contents of, or adding or deleting DIDs from, DOD-STD-2167A. Because existing DID format and content requirements preclude using new terminologies and exclude certain valuable categories of information, strategies which enable their inclusion should be considered (like adding appendices to the DIDs).

PDSS activities need access to software tests and reasons for design decisions arrived at during contractor software development. Additional considerations include:

- o restoring the Operational Concept Document as a document delivered to the contractor with the initial "A" Specification,
- o putting all requirements traceability in a traceability document for easy reference during PDSS,
- o including a DID Index which is keyed to all DID paragraph headings for easy reference whenever changes are made to any deliverable document based on a DOD-STD-2167A DID,
- o more realistic schedules for reviews,
- o use of incremental reviews,
- o consideration of ways to present information being reviewed so that it is easy to understand and not too voluminous (e.g., Critical Design Review), and
- o examining the software product evaluation in the standard for completeness and clarity.

It was strongly recommended that the JLC should stress that the software product evaluation activity is continuous, not discrete.

2.3 PANEL III: DOD-STD-2168: LESSONS LEARNED/ISSUES

2.3.1 Quality Indicators

DOD-STD-2168 was developed as a companion document to DOD-STD-2167A and supersedes MIL-S-52779A (Software Quality Assurance Program). DOD-STD-2168 establishes the requirements for specifying the contractor's software quality program. DOD-STD-2168 was developed with the objective of assuring the quality of (a) the deliverable software and its documentation, and (b) non-deliverable software. Non-deliverable software includes software used in the automated manufacturing of deliverable software or in the qualification or acceptance of deliverable software. In the three years since its issue, DOD-STD-2168 and its Software Quality Program Plan DID have been specified on a range of MCCR software development projects. SA-I, the first JLC workshop to review the standard, documented benefits and shortfalls of the standard and laid the groundwork for revision A of DOD-STD-2168. It was recommended that a requirement be added to DOD-STD-2168 that the software quality program include an effective measurement process using metrics and indicators. Metrics, indicators, and measurement definitions should also be added to DOD-STD-2168.

2.3.2 Consolidation into DOD-STD-2167A

Software quality is an essential component of software engineering. Software engineering is not defined in DOD-STD-2167A or DOD-STD-2168, nor is there recognition of the role of software quality in software development. DOD-STD-2167A and DOD-STD-2168 are perceived to have separate quality functions. DOD-STD-2167A requires those activities necessary to build in and design in quality. DOD-STD-2168 is interpreted to be a compliance/checking document. Overlapping and potentially conflicting requirements exist in DOD-STD-2167A and DOD-STD-2168 in the areas of evaluation records, document evaluations, corrective actions and software qualifications. A task to identify and recommend corrections to these overlapping/conflicting requirements should be a high priority effort.

2.3.3 Nomenclature Clarification

Clarification of the relationships between the development and engineering activities relating to software quality is needed. A corresponding clarification of the roles of these activities in the specified standards is also needed. A requirement is needed to evaluate the adequacy of products and processes. An additional

requirement is needed in DOD-STD-2168 that the criteria for all required evaluations be defined as part of the software quality program.

Paragraph 5.3.2 of DOD-STD-2168 needs to be expanded to address, and provide evaluation of, the full range of software engineering and development processes. MIL-HDBK-286 needs to be updated to reflect the corporate quality program for source selection to recognize the potential existence of the contractor's corporate quality program. DOD-STD-2168 should be reviewed to clarify the ambiguity between the corporate quality program and the contractor's quality program applied to a specific contract.

A requirement should be added to have management review of the implementation of the software quality program. This will ensure application of this requirement to each instance of the software quality program. The scope of the Software Test Plan must be expanded to include the requirement to describe the planning for the entire software test program, and to also include the test strategy. The Software Quality Program Plan should be revised to include a definition and description of the tasks to be performed. The Software Development Plan (SDP) should be revised to include a definition and description of the tasks to be performed for product evaluations.

The developmental configuration and product baseline needs to be expanded to include all deliverable software engineering and test documentation. The revision of DOD-STD-2167A should split the software design document into two software documents, a top level design and a detailed design description per the original concept. The Software Test Description should be split into two separate documents, one for test descriptions and one for test procedures. A DID should be added to DOD-STD-2167A for development of a Database Design Document. Software Configuration Management (CM) coverage should be removed from the Software Development Plan and placed into a Software CM Plan. DOD-STD-2168 should be revised to include a requirement for creating and updating a software quality program schedule. The DOD-STD-2167A Software Design Plan DID should be revised to accommodate acquisitions that include multiple Computer Software Configuration Items of multiple complexities with description of the development strategy and approach. DOD-STD-2168 should be revised to delete reference to MIL-STD-1520 (Corrective Action and Disposition System for Nonconforming Material).

2.4 PANEL IV: COMPUTER SECURITY/SOFTWARE INTEGRITY

2.4.1 Policies and Procedures

It was the decision of Panel IV to address the basic notion of computer security, namely that of reducing operational risks to computer support missions. The recommendations of Panel IV are

therefore not limited to those DoD systems to which the Trusted Computer System Evaluation Criteria (TCSEC) specifically applies, but apply to all DoD applications where the probability of a catastrophic flaw must be minimized or where failure of the computer system to operate as intended cannot be tolerated.

Panel IV endorses a previous recommendation from the Orlando II Workshop which stated that: "The Services must have an organic capability to evaluate systems against trusted computing criteria and certify them for the accreditation process. (The certification process provided by the National Computer Security Center (NCSC) takes too long.)" However, Panel IV notes that there is considerable confusion surrounding the use of the terms "evaluate", "certify", and "accredit". Therefore, Panel IV suggests that the recommendation be restated as: "The Services must have an organic capability to analyze systems against trusted computing criteria." Panel IV also endorsed the Orlando II Recommendation which stated: "The JLC should expedite the completion and release of standard language regarding security requirements for inclusion in contracts and SOW".

Security requirements must be defined early in the design process and be satisfied by the fielded system. This implies that designers and PMs be apprised of the need for this aspect of system design, and be directed to take the appropriate actions. This, in turn, implies the existence of policy which accomplishes the following: highlights the need for, and high-level concern about, the problem of identifying security requirements as part of system design and analysis; mandates action; justifies the expenditure of effort and resources toward this end. In the case of the identification of security requirements, it is important to understand that security is as much a system requirement as performance, size, and weight. Security requirements should not be identified and enumerated in a vacuum. Security should not be viewed as a single requirement. Security leads to many different requirements (e.g., access control, authentication).

The JLC should request that DODD-5200.28 (Security Requirements for Automated Information Systems) (or some other appropriate high-level directive) be amended to recognize the broad diversity of security issues, and to mandate the identification of security requirements as an integral part of the system design. In addition, DOD-STD-2167A needs to be amended to provide contractual vehicles (e.g., DIDs, Contract Data Requirements Lists) for specifying security attributes in system acquisitions.

This panel recommended that a set of procedures be established to ensure the uniform identification of security requirements in full coordination of all other system requirements. One approach to requirements identification uses a methodology, which has been applied to a number of mission critical systems. This methodology contains the following steps: security operational concept,

preliminary architecture, risk assessment, select security safeguards, and iterative process.

One recommended approach to providing the PM with the security experience needed to execute the methodology is the establishment of a security working group at the earliest stage in the process. It is also important to identify the Designated Approving Authority (DAA) at the beginning of the process. Ultimately, the DAA must accredit the system to process classified information. This early involvement will help to ensure that implementation trade-offs will be accreditable. Once policy has been established, PMs need assistance in drafting the contractual vehicles to assure that security requirements are included in the procurement process. In this area, this panel recommended that the JLC distribute the DoD "Procurement Guidelines," written by Grumman Corporation, to all system acquisition offices.

This panel recommended that the JLC establish accreditation policy and procedures across Services and to define procedures to reuse certification and accreditation results when possible. The processes should include plans for acquiring the resources to implement them, education and training for personnel with certification and accreditation responsibilities, transfer of responsibility, and proficiency standards for those personnel. Implementation of the following recommendations should be performed by the Computer Security Implementation Management Panel under the Joint Commanders Group on Communications-Electronics: develop standard certification and accreditation processes, define applicability of regulations, mandate process applicability, and develop additional guidance.

2.4.2 Specific Training

Panel IV enthusiastically endorsed the Orlando II Recommendation that the "...JPCG-CRM develop and coordinate a security awareness and training program for Project Managers and PDSS operational personnel." There is an urgent need by PMs at all levels for increased security awareness and a security training program. Experience has indicated that the few initial attempts at providing such training have been very well received. As a result of this Orlando II Recommendation, a course was developed entitled "Computer Security Answers for Acquisition Managers" for the National Security Agency (NSA). Additionally, a "Program Manager's Guide to Computer Security" was developed for NSA. These two efforts have not produced a continuing development program in this field. This program of training should be available to all DoD components, and it should be supported through a principal training element in DoD.

A requirement should be established early for technical security education throughout the life cycle of MCCR system developments. Technical security education and integration of security

engineering into the system life cycle process are not traditionally recognized as important aspects of developing secure systems. There are no requirements to educate management and technical personnel on the impacts of security to the system development. Security policy and requirements must be clearly communicated in terms of management support, technical activities and trade-off approaches for accomplishing a secure system that satisfies mission needs. Poor communication, lack of understanding, isolation of security engineering teams, de-emphasis of security and ignorance of the role of security in the overall system can lead to serious failure, accreditation non-compliance or excessive costs.

An awareness of computer security technology, ethics, and information security requirements should be developed in universities, military academies, and other educational organizations. Security awareness cannot start when someone has the job to develop a secure system. The principles of security, trust, and ethics must be taught to the user communities early in their development. The need to educate users in the technology and its application cannot start too early. The above educational institutions should develop courses that address computer security requirements, risk analysis, trusted systems, and ethics.

2.4.3 Security Question

Experience has shown that systems which must be developed under DOD-STD-2167A and which must satisfy security requirements often are not able to satisfy both sets of requirements. This leads to system acquisitions which satisfy the contractually mandated DOD-STD-2167A requirements but which fall short of satisfying critical security requirements. Ultimately such systems provide less assurance and are difficult to certify. This panel therefore believes that it is imperative that PMs be provided the contractual vehicle to require an integrated development approach to building secure systems under DOD-STD-2167A.

Panel IV strongly endorses the Orlando II recommendation which states: "The JLC should establish a committee to develop changes to DOD-STD-2167 that incorporate security requirements as an integral part of a systems development life cycle." The standard must include specific Service requirements as well as National Computer Security Center requirements; it must provide DIDs to detail the required deliverables; and it should be augmented by a guidebook for application of the security standards. The basis for this standard should proceed from an appropriate modification to DODD-5000.1 (Policies Governing Defense Acquisition) and DODI-5000.2 (Defense Acquisition Management Policies and Procedures).

A number of efforts have been undertaken in this area and should be used as a basis for immediate action. In addition, members of Panel IV met with members of Panel II (DOD-STD-2167A and DIDs)

during the SA-I workshop to discuss the insertion of security policies and procedures into DOD-STD-2167A.

2.5 PANEL V: CONFIGURATION MANAGEMENT

2.5.1 Detailed Recommendations

The Defense Quality Standardization Office and the JPCG-CRM PDSS Subgroup are jointly sponsoring the development of a military standard and associated handbook covering all aspects of CM in DoD. The project will provide, for the first time, a comprehensive, top-down approach to CM as it relates to all aspects of DoD weapon system acquisition and life cycle support, including hardware, firmware, and software configuration items. Although two new documents will be developed, 11 existing documents will be deleted from the Government standards inventory. Additional objectives of the project are to identify conflicts and other problem areas among the current versions of associated standards as they relate to CM, particularly MIL-STD-490A (Specification Practices), MIL-STD-499A (Engineering Management), and DOD-STD-2167A, and to prepare formal recommendations for resolving these problems.

MIL-STD-973 (Configuration Management) is scheduled to be published in late summer 1991. The 25 January 1991 draft version of MIL-STD-973 was reviewed by Panel V. Recommendations of the panel include:

- the standard should ensure definition and coverage of computer software and related items;
- a requirement should be added for the accompanying handbook to provide guidance to Government agencies on the application of the standard;
- DOD-STD-1679A (Software Development) defined Software Change Proposals which should be attached to Engineering Change Proposals;
- computer software physical characteristics should be redefined;
- the standard should be restructured for readability;
- CM discipline requirements should be added for Non-Development Items (NDI), Privately Developed Items, Commercial-Off-the-Shelf (COTS) Items, prototypes and development configurations;
- implementation guidance should be provided;
- computer software CM status accounting data elements should be added;
- definitions should be changed or added as needed; and
- concerns should be addressed relative to MIL-STD-973 and MIL-STD-499A publication versus MIL-STD-1521B (Technical Reviews and Audits for Systems, Equipments, and Computer Programs) supersession.

2.6 PANEL VI: SOFTWARE REUSABILITY

2.6.1 Broadened Definition Required

It has been said that the surest way to avoid software development costs is to avoid developing software--just reuse existing software. Monterey II was the previous formal JLC initiative to address software reuse issues. That initiative provided the framework to establish DOD-STD-2167A and to accept the concept of a single high order language as the cornerstone to achieving the reusable asset goal. Reuse is perceived by many as the use of a software asset in an application other than that for which it was originally developed. The panel believes that this definition is too narrow. It precludes the concept of expressly developing software assets for use in a number of applications or a family of such systems within a domain.

Panel VI reaffirms that reuse of existing software assets is one of the most promising approaches to improving software quality, reducing software cost, and shortening development schedules. However, many studies and recommendations over the past ten years have failed to significantly change the amount of reuse happening in DoD software. In part, the problem is that DoD has failed to act on the recommendations of previous studies. Software reusability is a difficult process to implement within the DoD structure, primarily because there are no effective incentives which encourage the creation of the types of assets necessary for reuse. Software reusability concepts to date do not represent a revolutionary technological advance but rather an incremental, methodological, and evolutionary advance of an existing process. The panel foresees that the degree of community consensus on a common architecture in a given domain will broaden over time. In the near-term, there are instances of domain-specific architecture and assets within companies and organizations. There are also instances of licensable libraries. If the JLC acts upon the recommendations in this report, then in the mid-term, architecture-based reuse could be achieved within a Program Executive Officer's (PEO's) area and as the common way of doing business within a company. The panel foresees limited mid-term intercompany sharing of proprietary architectures and assets in instances of prime and subcontractor relationships and consortia. With strong DoD support, the panel envisions broad-based institutionalization of architecture-based reuse within ten years. This can serve as the basis for a software components industry. The industry would consist of licensable COTS libraries for assets which have dual usage (support both DoD and commercial requirements) and self-sustaining, domain-specific libraries of DoD-specific NDI components which operate as a service industry.

2.6.2 Incentives/Roadblocks

Previous studies of how DoD can derive more benefit from software reuse have consistently identified lack of incentives as a barrier to reuse. After reviewing the studies above, discussions at SA-I reached a consensus that lack of incentives to create reusable software is a bigger problem than lack of incentives to use reusable software assets when they are available. Both Government and contract PMs have well understood incentives to reduce cost, shorten schedules, and improve quality for the project at hand. Given confidence in its quality, performance, and the suitability of its interfaces, PMs will reuse software to help reach these goals. On the other hand, a substantial effort is required to create reusable software assets. In addition to normal software engineering practices, the developers must identify the range of possible contexts and applications where the software will be applied, and make tradeoffs between generality (which expands the domain of applicability) and efficiency in each specific application. This indicates that a substantial investment is required for reuse.

Three scenarios were identified where reuse has been widely and successfully employed as an integral part of the software engineering process:

- where there are multiproject organizations which produce, maintain, and evolve a family of related products;
- where there is a technical director or system architect who takes responsibility for seeking out opportunities for commonality and reuse among products of the same generation and between products of successive generations; and
- where there is use of COTS tools and components.

Panel VI established a series of high-level recommendations to address reuse barriers. Reuse is currently treated as a separate process from software engineering. Reuse should be integrated completely into the life-cycle. There is a lack of effective incentives to create, support, and promote use of reusable software assets. The JLC should request the Under Secretary of Defense (Acquisition) (USD(A)) to create groups that will serve as a mechanism to facilitate the sharing of reusable software among programs which are in the same domain but report to different PEOs. In addition, the JLC should request the Defense Advanced Research Projects Agency (DARPA) to explore the increased use of domain specific software architecture interface standards as a mechanism which gives companies confidence that assets complying with these interface standards will have a market. The JLC should also request that USD(A) set up a program to recognize original authors (organization and individuals) of software assets each time they are actually reused.

Recoupment policy currently inhibits contractors from investing in dual use (Government and commercial) reusable software which would be beneficial to DoD. The JLC should request that USD(A) promulgate policy mandating up-front negotiation of recoupment in software acquisitions, and in the absence of such negotiations, contracts should not contain a recoupment clause. Also, the JLC should begin a dialogue with the National Security Industrial Association (NSIA) to support the association's project to establish an equitable recoupment policy and to request NSIA to broaden their investigation to include industry recoupment. Concerns about confidence and lack of quality in reusable assets also inhibit reuse. The panel believes that in the long term the most effective form of asset reuse will be "planned, black-box" reuse rather than opportunistic reuse with repeated ad hoc modification.

Legal issues regarding ownership and liability present barriers to reusing software assets. The JLC should request that USD(A) direct the Services to train acquisition personnel in understanding the options available for acquisition of software assets and the situations that best serve the Government. The panel also discussed the need for development of a centralized catalog of assets. The panel does not recommend the establishment of, or support for, a centralized catalog or library to support reuse at this time.

2.6.3 Implementation Responsibilities

The JLC should foster integration of domain analysis with the early system development process:

- requesting the Service Acquisition Executives (SAEs) to issue policy that makes the PEOs responsible for domain analysis;
- requiring the PEOs to review, advise, and be part of the team defining new user requirements; and
- assuring that the representation of a family of requirements is used to verify all new requirements posed by the user.

The JLC should request that SAEs support Service technology base work in the area of domain analysis, particularly in application of domain analysis to the requirements definition phase, and in creation of high level reusable assets (such as software architectures) in their Critical Technology Plans.

The JLC should encourage use of COTS. A guidebook should be developed for contracting officers and PMs which clarifies the nuances of how to license COTS components for use in Mission Critical Computer Systems. The JLC should request that USD(A) direct the Services to more thoroughly train acquisition personnel in the negotiation of software licenses for COTS so as to ensure

that only fully qualified personnel represent the Government in such negotiations. The JLC should encourage studies to identify standards necessary for reuse. In addition, industry standards groups should pursue definition and evolution of these standards. USD(A) should be requested by the JLC to ensure that providing reuse incentives becomes the specific responsibility of PEOs.

To the extent legal risks of reusing software are the same as using first-use software, perceptions of greater risk can be mitigated through education. Therefore, the JLC should request that USD(A) direct the Services to train acquisition personnel in understanding the options available for acquisition of software assets and the situations that best serve the Government. SAEs should be requested by the JLC to have the Services sponsor pilot projects to explore existing domain analysis methods as part of their technology base programs. The JLC should also request that USD(A) issue policy supporting the use of prototyping in all life-cycle phases to determine which existing assets can be used to implement proposed requirements.

USD(A) should be requested by the JLC to mandate inclusion of COTS component/asset evaluation with experimental prototyping in the requirements definition and architecture definition phases of program plans. The JLC should request that USD(A) direct the development of an approach to capture the lineage and track record of reusable assets as a way of promoting confidence. The JLC should request that SAEs undertake a near-term effort to establish for one or more initial domains, a set of "dimensions" to describe the semantics of the components in one or more domains. In addition, the JLC should request that DARPA and the Services initiate a technology action plan to define a means for precise software asset description (equivalent to a hardware specification sheet). Also, the JLC should request that industry associations establish a working group to develop near-term common descriptions for reusable components, forms, and interfaces.

The USD(A) should be requested by the JLC to issue a policy allowing contractor participation in early mission activity to perform domain analysis for families of systems. The Computer Software Management Subgroup of the JPCG-CRM should assess the potential impact of a need for asset descriptions on DOD-STD-2167A DIDs to ensure the DIDs would permit such description.

2.7 PANEL VII: ADA SECONDARY STANDARDS

2.7.1 Ada Bindings

The panel discussed standard interfaces and Ada bindings. In general, a standard interface defines a set of services provided to an application. In order to use these interface services with Ada, there must be a definition of the interface expressed in terms of the Ada language. This definition is called an "Ada

binding" to the interface. DoD must participate in the definition of standard interfaces and Ada bindings to those interfaces to ensure that the standard supports DoD needs since industry standards are often developed from commercial practices which differ from DoD practices in some way.

The current demand for Ada bindings was discussed. DoD has emphasized the use of COTS products and standards in acquisitions. This emphasis has increased the demand for implementations of Ada bindings to these products and standards. For many of these products and standards, there are no Ada implementations available. Developing and implementing an Ada binding requires a substantial amount of work. A PM may find that it is neither cost-effective nor schedule-effective to fund that development from within the project. This often leads the PM to request a waiver from Ada use.

To support DoD's Ada policy and to amortize the cost of producing these interfaces, it makes sense to fund the development of Ada bindings to commonly used COTS products and standards. Based on the experience of the panel members, several "high-payoff" interfaces can be nominated, with the recommendation that the JLC fund the development of these standards, Ada bindings, and implementations. The development of high quality Ada bindings requires substantial resources. The return on investment for a well-engineered binding is significant, as both implementing and using the binding becomes much easier and less error-prone. The best software engineers must be actively involved in developing Ada interfaces. A long-term commitment to the task is critical.

Public review is also critical to the success of a binding. But it is costly, particularly in terms of development schedule. During the review of a binding, a prototype implementation may decrease the effort involved in development of a production-quality version. Once review is complete, vendor and third-party software houses can build upon the benefits of the review and prototype. Coordination of all parties' energies leads to a lower-risk Ada interface technology; therefore, the DoD will benefit by investing in the review and prototype process.

Panel VII recommends the creation of a DoD-wide Ada Interface Technology Initiative (AITI), involving the JLC, the Service PEOs, and the DoD Consolidated Software Initiative activities (Software Engineering Institute (SEI), Software Technology for Adaptable and Reliable Systems (STARS), and the Ada Joint Program Office (AJPO)). Funding for this activity should be included as a separate budget item in the DoD Consolidated Software Initiative budget.

The AITI should produce application profiles for DoD applications. An application profile is a coherent set of standards and Ada interfaces for a specific applications domain. Responsibility for

the creation of these profiles is to reside with the PEOs, with assistance from the DoD Consolidated Software Initiative. As the PEOs are responsible for specific application domains, they must be at the center of profile creation.

The panel further recommends that funding should be allocated immediately to fund development of the following interfaces (in priority order):

1. X-Windows, especially Open System Foundations Motif
2. Standard Query Language (SQL) and the SQL Ada Module Extension
3. Portable Operating System Interface for UNIX (POSIX) Computer Environments
4. Government Open Systems Interconnection Profiles, including Transmission Control Protocol and INTERNET Protocol
5. Information System Services support, especially decimal arithmetic
6. Numerical Functions
7. Graphics interfaces such as Graphics Kernel Systems (GKS), Graphics Interfaces Programmers Hierarchial Interactive Graphics Systems (PHIGS), and PHIGS Extension for X
8. Ada/Atlas Testing Language

2.7.2 Specific Application

A profile is a set of coherent standards that provides interfaces, services and supporting formats for interoperability and portability of applications, data and people for a specific applications domain. For example, a profile for C^3 systems might specify the Institute for Electrical and Electronic Engineers (IEEE) POSIX as its operating system interface, SQL for its database management interface, GKS for its graphics interface, and X-Windows for its user interface. A profile for Information Systems might include IEEE POSIX, SQL, and X-Windows, but not GKS, as it has no need for graphics. The Information Systems profile would also include bindings for decimal arithmetic and indexed files.

Since a profile is a set of services tailored to a specific domain, the set should be "necessary" but need not be "complete". There should be a clear need for the specified service within the given domain. Given a set of required services, the next step is to select a standard to fulfill that requirement. The selected standard should meet the overall needs of the applications domain. For instance, an operating system for a embedded real-time domain should support precise timing and synchronization events. Given the changing state of the standards world, it is possible that some requirements of a given profile are not satisfied by an existing standard. In this case, the profile developer can either

acknowledge the unfilled requirement, or can start a standardization activity to fill that requirement. Like the development of Ada bindings, profile development also requires high-quality engineers with experience in the applications domain, Ada, and in the component standards.

The next step in developing a profile is to study the interaction of the selected standards. This is the key systems engineering effort in developing a profile. The profile will likely constrain and augment its component standards to support this integration.

The panel recommended that the JLC initiate a process leading to the establishment of application domain profiles, to be subsequently applied to JLC software management. This process should lead to the specification of operational profiles, should involve PEOs, and should be supported by those involved in the DoD Consolidated Software Initiative (STARS, SEI, AJPO). The rationale for the definition and use of application domain profiles, and the specific involvement of these DoD agencies, is clear. The current state-of-the-practice in DoD Ada interface technology has been fragmented and a high-level, DoD-wide initiative will counter this fragmentation.

A profile should be application domain oriented, and should support the requirements for Ada application development in that domain. In this way, many DoD programs will be able to leverage the investment made in the definition of applicable profiles. Leadership for the definition of the profiles must fall on the PEOs.

2.7.3 Acceleration of Ada Technology Insertion

The JLC should extend DoD policy to require that state-of-the-art computer technology is compatible with Ada. This is particularly important in the earlier, introductory stages of computer technology development. This will lead to reduced risk in mature systems and higher predictability in software management. An important tenet of a DoD Ada interface technology initiative will be a plan to manage the insertion of such technology in long range programs, such as advanced research. The current state of practice of demonstrating over-the-horizon technologies without Ada leads to a long lag between demonstration and the ability to apply Ada in those technologies. The goal is to ensure that Ada shall be no more difficult to use than any other language in DoD-sponsored technologies.

3.0 SUMMARY

The central theme of the SA-I Software Workshop was "DoD Software for the 1990s." The primary purpose of the SA-I Software Workshop was to review key issues, common to the Services and industry, which are related to the acquisition and development of MCCR. The

seven panels were also tasked with making specific recommendations to improve their topic area. The panels met their challenge by intensively reviewing their assigned topics and proposing 137 recommendations for improvement. These recommendations could ensure that DoD software development in the 1990s will be more measurable, standardized, secure, configuration managed, reusable, and Ada-oriented than in the past. The task now before the JPCG-CRM is to address these recommendations.

4.0 LIST OF ACRONYMS

AFR	Air Force Regulation
AITI	Ada Interface Technology Initiative
AJPO	Ada Joint Program Office
AR	Army Regulation
CM	Configuration Management
COTS	Commercial Off-The-Shelf
CSM	Computer Software Management
DAA	Designated Approving Authority
DARPA	Defense Advanced Research Projects Agency
DIDs	Data Item Descriptions
DoD	Department of Defense
DODD	Department of Defense Directive
DODI	Department of Defense Instruction
GKS	Graphics Kernel Systems
HDBK	Handbook
IEEE	Institute for Electrical and Electronic Engineers
JLC	Joint Logistics Commanders
JPCG-CRM	Joint Policy Coordinating Group for Computer Resources Management
MCCR	Mission Critical Computer Resources
MIL	Military
NCSC	National Computer Security Center
NDI	Non-Developmental Item
NSA	National Security Agency
NSIA	National Security Industrial Association
PDSS	Post Deployment Software Support
PEO	Program Executive Officer
PHIGS	Programmers Hierarchial Interactive Graphics Systems
PM	Program/Project Manager
POSIX	Portable Operating System Interface for UNIX
SAEs	Service Acquisition Executives
SECNAVINST	Secretary of the Navy Instruction
SEI	Software Engineering Institute
SOW	Statement of Work
STARS	Software Technology for Adaptable and Reliable Systems
STD	Software Test Description
TCSEC	Trusted Computer System Evaluation Criteria
USD(A)	Under Secretary of Defense (Acquisition)

(INTENTIONAL BLANK)